Chapter 1

Iterative Methods for Linear Systems

1.1 Basic Concepts and Stationary Iterative Methods

1.1.1 Review and Notations

Notation 1.1.1. We will write linear equations as

$$Ax = b \tag{1.1}$$

where A is a nonsingular $N \times N$ matrix, $b \in \mathbb{R}^N$ is given, and

$$x^* = A^{-1}b \in \mathbb{R}^N$$

is to be found.

Notation 1.1.2. Throughout this chapter, x will denote a potential solution and $\{x_k\}_{k>0}$ the sequence of iterates. We will denote the ith component of a vector x by $(x)_i$ and the ith component of x_k by $(x_k)_i$.

Notation 1.1.3. In this chapter, $\|\cdot\|$ will denote a norm on $\mathbb{R}^N$ as well as the *induced matrix norm*. We will denote the *condition number* of A relative to the norm $\|\cdot\|$ by

$$\kappa(A) = \|A\|\|A^{-1}\|$$

where $\kappa(A)$ is understood to be infinite if A is singular.

Definition 1.1.4. Let $\|\cdot\|$ be a norm on $\mathbb{R}^N$. The *induced matrix norm* of an $N \times N$ matrix A is defined by

$$\|A\| = \max_{\|x\|=1} \|Ax\|.$$

Proposition 1.1.5. Induced norms have the important property that $\|Ax\| \leq \|A\|\|x\|$

Definition 1.1.6. The *error* of the *iterative methods* is

$$e = x - x^*$$

Definition 1.1.7. The *residual* of a potential solution x to (1.1) is

$$r = b - Ax$$

Lemma 1.1.8. Let $b,\ x,\ x_0 \in \mathbb{R}^N$. Let A be nonsingular and let $x^* = A^{-1}b$.

$$\frac{\|e\|}{\|e_0\|} \leq \kappa(A)\frac{\|r\|}{\|r_0\|}. \tag{1.2}$$

Proof. Since

$$r = b - Ax = -Ae$$

we have

$$\|e\| = \|A^{-1}Ae\| \leq \|A^{-1}\|\|Ae\| = \|A^{-1}\|\|r\|$$

and

$$\|r_0\| = \|Ae_0\| \leq \|A\|\|e_0\|.$$

Hence

$$\frac{\|e\|}{\|e_0\|} \leq \frac{\|A^{-1}\|r\|}{\|A\|^{-1}\|r_0\|} = \kappa(A)\frac{\|r\|}{\|r_0\|}. \qquad \square$$

Remark 1.1.1. Most iterative methods terminate when the residual is sufficiently small. One termination criterion is

$$\frac{\|r_k\|}{\|r_0\|} < \tau, \tag{1.3}$$

It depends on the initial iterate and may result in unnecessary work when the initial iterate is good and a poor result when the initial iterate is far from the solution. For this reason we prefer to terminate the iteration when

$$\frac{\|r_k\|}{\|b\|} < \tau. \tag{1.4}$$

1.1.2 The Banach Lemma and approximate inverses

Definition 1.1.9. The *Richardson iteration* for solving a linear system (1.1) is an iteration of the form

$$x_{k+1} = (I - A)x_k + b. \tag{1.5}$$

We will discuss more general methods in which $\{x_k\}$ is given by

$$x_{k+1} = Mx_k + c. \tag{1.6}$$

Definition 1.1.10. Iterative methods of form (1.5) and (1.6) are called *stationary iterative methods.*

Remark 1.1.2. The Richardson iteration is equivalent to the fixed-point iteration for $f(x) = (I - A)x + b$.

Remark 1.1.3. The transition from x_k to x_{k+1} does not depend on the history of the iteration. The Krylov methods discussed in next two sections are not stationary iterative methods.

Lemma 1.1.11. If M is an $N \times N$ matrix with $\|M\| < 1$ then $I - M$ is *nonsingular* and

$$\|(I-M)^{-1}\| \leq \frac{1}{1-\|M\|}. \tag{1.7}$$

Proof. We will show that $I - M$ is nonsingular and that (1.7) holds by showing that the series

$$\sum_{l=0}^{\infty} M^l = (I-M)^{-1}.$$

The partial sums

$$S_k = \sum_{l=0}^{k} M^l$$

form a Cauchy sequence in $\mathbb{R}^{N\times N}$. To see this, note that for all $m > k$,

$$\|S_k - S_m\| \leq \sum_{l=k+1}^{m} \|M\|^l = \|M\|^{k+1}\left(\frac{1-\|M\|^{m-k}}{1-\|M\|}\right) \to 0$$

as $m, k \to \infty$. Hence the sequence S_k converges, say to S. Since $MS_k + I = S_{k+1}$, we have $MS + I = S$.

$$\|(I-M)^{-1}\| \leq \sum_{l=0}^{\infty} \|M\|^l = (1-\|M\|)^{-1},$$

so we prove that $I - M$ is nonsingular and $S = (I - M)^{-1}$. □

Corollary 1.1.12. If $\|M\| < 1$ then the iteration (1.6) converges to $x = (I - M)^{-1}c$ for all initial iterates x_0.

Remark 1.1.4. A consequence of Corollary 1.1.12 is that Richardson iteration (1.5) will converge if $\|I - A\| < 1$. It is sometimes possible to *precondition* a linear equation by multiplying both sides of (1.1) by a matrix B

$$BAx = Bb$$

so that convergence of iterative methods is improved.

Definition 1.1.13. B is an *approximate inverse* of A if $\|I - BA\| < 1$.

Remark 1.1.5. The approximate inverse allows us to apply the Banach lemma and its corollary to the preconditioned Richardson iteration.

Theorem 1.1.14. If A and B are $N \times N$ matrices and B is an approximate inverse of A, then A and B are both nonsingular and

$$\|A^{-1}\| \leq \frac{\|B\|}{1-\|I-BA\|}, \quad \|B^{-1}\| \leq \frac{\|A\|}{1-\|I-BA\|}, \tag{1.8}$$

and

$$\|A^{-1} - B\| \leq \frac{\|B\|\|I-BA\|}{1-\|I-BA\|}, \; \|A - B^{-1}\| \leq \frac{\|A\|\|I-BA\|}{1-\|I-BA\|}. \tag{1.9}$$

Proof. Let $M = I - BA$. By Lemma 1.1.11 $I - M = I - (I - BA) = BA$ is nonsingular. Hence both A and B are nonsingular. By (1.7)

$$\|A^{-1}B^{-1}\| = \|(I-M)^{-1}\| \leq \frac{1}{1-\|M\|} = \frac{1}{1-\|I-BA\|}. \tag{1.10}$$

Since $A^{-1} = (I - M)^{-1}B$, inequality (1.10) implies the first part of (1.8). The second part follows in a similar way from $B^{-1} = A(I - M)^{-1}$.

To complete the proof note that

$$A^{-1} - B = (I - BA)A^{-1}, \; A - B^{-1} = -B^{-1}(I - BA).$$

and use (1.8). □

Remark 1.1.6. Richardson iteration, preconditioned with approximate inversion, has the form

$$x_{k+1} = (I - BA)x_k + Bb. \tag{1.11}$$

If the norm of $I - BA$ is small, then not only will the iteration converge rapidly, but, as Lemma 1.1.8 indicates, termination decisions based on the preconditioned residual $Bb - BAx$ will better reflect the actual error.

Remark 1.1.7. There are many approaches to precondition the linear equations. For example, multigrid methods can also be interpreted in this light.

1.1.3 The spectral radius

Example 1.1.15. The norm of M could be small in some norms and quite large in others. For example, letting

$$M = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.25 \end{bmatrix}$$

$\|M\|_1 = 0.75, \|M\|_\infty = 2$

Remark 1.1.8. The analysis in 1.1.2 related convergence of the iteration (1.6) to the norm of the matrix M. However the norm of M could be small in some norms and quite large in others.

Notation 1.1.16. We let $\sigma(A)$ denote the set of eigenvalues of A.

Definition 1.1.17. The *spectral radius* of an $N \times N$ matrix A is

$$\rho(A) = \max_{\lambda\in\sigma(A)} |\lambda| = \lim_{n\to\infty} \|A^n\|^{1/n} \tag{1.12}$$

Lemma 1.1.18. For any induced matrix norm, we have

$$\rho(A) \leq \|A\|. \tag{1.13}$$

Theorem 1.1.19. Let A be an $N \times N$ matrix. Then for any $\epsilon > 0$, there is a norm $\|\cdot\|$ on $\mathbb{R}^N$ such that

$$\rho(A) > \|A\| - \epsilon.$$

Theorem 1.1.20. Let M be an $N \times N$ matrix. The iteration (1.6) converges for all $c \in \mathbb{R}^N$ if and only if $\rho(M) < 1$.

Theorem 1.1.21. If $\rho(M) \geq 1$ then there are some x_0 and c such that the iteration (1.6) fails to converge.

Remark 1.1.9. A consequence of Theorem 1.1.19, 1.1.20 and Lemma 1.1.11 is a characterization of convergent stationary iterative methods.

1.1.4 Matrix splittings and classical stationary iterative methods

Definition 1.1.22. Methods such as Jacobi, Gauss-Seidel, and successive overrelaxation (SOR) iteration are based on *splittings* of A of the form

$$A = A_1 + A_2,$$

where A_1 is a nonsingular matrix constructed so that equations with A_1 as coefficient matrix are easy to solve. Then $Ax = b$ is converted to the fixed-point problem

$$x = A_1^{-1}(b - A_2 x).$$

Remark 1.1.10. The analysis of the method is based on an estimation of the spectral radius of the iteration matrix $M = A_1^{-1}A_2$.

Example 1.1.23. We consider the Jacobi iteration that uses the splitting

$$A_1 = D,\ A_2 = L + U,$$

where D is the diagonal of A, L and U are the (strict) lower and upper triangular parts. This leads to the iteration matrix

$$M_{JAC} = -D^{-1}(L + U).$$

Letting $(x_k)_i$ denote the ith component of the kth iterate we can express Jacobi iteration concretely as

$$(x_{k+1})_i = a_{ii}^{-1}\left(b_i - \sum_{j \neq i} a_{ij}(x_k)_j\right). \tag{1.14}$$

We will present the convergence result for Jacobi iterative method.

Theorem 1.1.24. Let A be an $N \times N$ matrix and assume that for all $1 \leq i \leq N$

$$0 < \sum_{j \neq i} |a_{ij}| < |a_{ii}|. \tag{1.15}$$

Then A is nonsingular and the Jacobi iteration (1.14) converges to $x^* = A^{-1}b$ for all b.

Proof. Note that the ith row sum of $M = M_{JAC}$ satisfies

$$\sum_{j=1}^{N} |m_{ij}| = \frac{\sum_{j \neq i} |a_{ij}|}{|a_{ii}|} < 1.$$

Hence $\|M_{JAC}\|_\infty < 1$ and the iteration converges to the unique solution of $x = Mx + D^{-1}b$. Also $I - M = D^{-1}A$ is nonsingular and therefore A is nonsingular. □

Example 1.1.25. Gauss-Seidel iteration overwrites the approximate solution with the new value as soon as it is computed. This results in the iteration

$$(x_{k+1})_i = a_{ii}^{-1}\left(b_i - \sum_{j<i} a_{ij}(x_{k+1})_j - \sum_{j>i} a_{ij}(x_k)_j\right)$$

the splitting

$$A_1 = D + L,\ A_2 = U,$$

and iteration matrix

$$M_{GS} = -(D + L)^{-1}U.$$

Remark 1.1.11. We can view these methods as a preconditioned Richardson iteration. For the Jacobi iteration, the approximate inverse

$$B_{JAC} = D^{-1}.$$

For Gauss-Seidel iteration,

$$B_{GS} = (D + L)^{-1}.$$

1.2 Conjugate Gradient Iteration

1.2.1 Krylov methods and the minimization property

Definition 1.2.1. Given a nonsingular $A \in \mathbb{C}^{N \times N}$ and $y \neq \mathbf{0} \in \mathbb{C}^N$, the nth *Krylov subspace* $\mathcal{K}_n(A, y)$ generated by A from y is

$$\mathcal{K}_n := \mathcal{K}_n(A, y) = \mathbf{span}(y, Ay, \ldots, A^{n-1}y).$$

Definition 1.2.2. A *Krylov space method* for solving a linear system $Ax = b$ is an iterative method starting from some initial approximation x_0, the corresponding residual $r_0 := b - Ax_0$ and generating for all or at least most n, until it possibly finds the exact solution, iterates x_n such that

$$x_n - x_0 = q_{n-1}(A)r_0 \in \mathcal{K}_n(A, r_0)$$

with a polynomial q_{n-1} of exact degree $n - 1$. For some n, x_n may not exist or q_{n-1} may have lower degree.

Example 1.2.3. The two such methods that we will discuss in depth, conjugate gradient and GMRES, minimize, at the kth iteration, some measure of error over the affine space

$$x_0 + \mathcal{K}_k,$$

where x_0 is the initial iterate and the kth Krylov subspace $\mathcal{K}_k$ is

$$\mathcal{K}_k = \mathcal{K}_k(A, r_0) = \mathbf{span}(r_0, Ar_0, \ldots, A^{k-1}r_0)$$

for $k \geq 1$.

Remark 1.2.1. Unlike the classical work, in the following sections, we will begin with a description of what the algorithm does and the consequences of the minimization property of the iterates. After that we describe termination criterion, performance, preconditioning, and at the very end, the implementation.

Definition 1.2.4. A linear system $Ax = b$ is called *symmetric positive definite (spd)* systems if A is symmetric and positive definite, i.e.,

$$A = A^T \text{ and } x^T Ax > 0 \text{ for all } x \neq 0$$

CG iteration is intended to solve symmetric positive definite (spd) systems.

Definition 1.2.5. Given a symmetric positive definite matrix A, we may define a norm by

$$\|x\|_A = \sqrt{x^T Ax} \tag{1.16}$$

which is called the A-norm.

Lemma 1.2.6. The kth iterate x_k of CG minimizes

$$\phi(x) = \frac{1}{2}x^T Ax - x^T b \tag{1.17}$$

over $x_0 + \mathcal{K}_k$. If $\phi(\tilde{x})$ is the minimal value in $\mathbb{R}^N$, then $\tilde{x} = x^* := A^{-1}b$.

Proof. $\phi(\tilde{x})$ is the minimal value, then

$$\nabla\phi(\tilde{x}) = A\tilde{x} - b = 0. \qquad \square$$

Lemma 1.2.7 (minimization property of CG iteration)**.** Let $\mathcal{S} \subset \mathbb{R}^N$. If x_k minimizes ϕ over $\mathcal{S}$ then x_k also minimizes $\|x^* - x\|_A = \|r\|_{A^{-1}}$ over $\mathcal{S}$.

Proof. Note that

$$\|x - x^*\|_A^2 = x^T Ax - x^T Ax^* - (x^*)^T Ax + (x^*)^T Ax^*.$$

Since A is symmetric and $Ax^* = b$,

$$-x^T Ax^* - (x^*)^T Ax = -2x^T Ax^* = -2x^T b.$$

Therefore

$$\|x - x^*\|_A^2 = 2\phi(x) + (x^*)^T Ax^*.$$

Since $(x^*)^T Ax^*$ is independent of x, minimizing ϕ is equivalent to minimizing $\|x - x^*\|_A^2$ and hence to minimizing $\|x - x^*\|_A$.

If $e = x - x^*$ then

$$\|e\|_A^2 = e^T Ae = (A(x-x^*))^T A^{-1}(A(x-x^*)) = \|b - Ax\|_{A^{-1}}^2$$

and hence the A-norm of the error is also A^{-1}-norm of the residual. $\square$

Remark 1.2.2. We will use this lemma in the particular case that $\mathcal{S} = x_0 + \mathcal{K}_k$ for some k.

1.2.2 Consequences of the minimization property

Lemma 1.2.8. In the kth step of CG iteration, we have

$$\|x^* - x_k\|_A = \min_{p\in\mathbf{P}_k, p(0)=1} \|p(A)(x^* - x_0)\|_A \tag{1.18}$$

where $\mathbf{P}_k$ denotes the set of polynomials of degree k.

Proof. Lemma 1.2.7 implies that since x_k minimizes ϕ over $x_0 + \mathcal{K}_k$

$$\|x^* - x_k\|_A \leq \|x^* - \omega\|_A \tag{1.19}$$

for all $\omega \in x_0 + \mathcal{K}_k$ can be written as

$$\omega = \sum_{j=0}^{k-1} \gamma_j A^j r_0 + x_0$$

for some coefficients $\{\gamma_j\}$, we can express $x^* - \omega$ as

$$x^* - \omega = x^* - x_0 - \sum_{j=0}^{k-1} \gamma_j A^j r_0.$$

Since $Ax^* = b$ we have

$$r_0 = b - Ax_0 = A(x^* - x_0)$$

and therefore

$$x^* - \omega = x^* - x_0 - \sum_{j=0}^{k-1} \gamma_j A^{j+1}(x^* - x_0) = p(A)(x^* - x_0)$$

where the polynomial $p(z)$ has degree k and satisfies $p(0) = 1$. Hence (1.18) is proved. $\square$

Lemma 1.2.9. In the kth step of CG iteration, we have

$$\frac{\|x^* - x_k\|_A}{\|x^* - x_0\|_A} = \min_{p\in\mathbf{P}_k, p(0)=1} \max_{z\in\sigma(A)} |p(z)| \tag{1.20}$$

where $\sigma(A)$ is the set of all eigenvalues of A.

Proof. The spectral theorem for spd matrices asserts that

$$A = U\Lambda U^T,$$

where U is an orthogonal matrix whose columns are the eigenvectors of A and Λ is a diagonal matrix with the positive eigenvalues of A on the diagonal. Since $UU^T = U^T U = I$ by orthogonality of U, we have

$$A^j = U\Lambda^j U^T.$$

Hence $p(A) = Up(\Lambda)U^T$. Define $A^{1/2} = U\Lambda^{1/2}U^T$ and note that

$$\|x\|_A^2 = x^T Ax = \|A^{1/2}x\|_2^2. \tag{1.21}$$

Hence, for any $x \in \mathbb{R}^N$,

$$\begin{aligned}\|p(A)x\|_A &= \|A^{1/2}p(A)x\|_2 \\ &\leq \|p(A)\|_2\|A^{1/2}x\|_2 = \|p(A)\|_2\|x\|_A.\end{aligned}$$

This, together with (1.18), (1.20) is proved. $\square$

Corollary 1.2.10. Let A be spd and let $\{x_k\}$ be the CG iterates. Let k be given and let $\overline{p}_k$ be any kth degree polynomial such that $\overline{p}_k(0) = 1$. Then

$$\frac{\|x_k - x^*\|_A}{\|x_0 - x^*\|_A} \leq \max_{z\in\sigma(A)} |\overline{p}_k(z)|. \tag{1.22}$$

Definition 1.2.11. The set of kth degree *residual polynomials* is

$$\mathcal{P}_k = \{p \mid p \text{ is a polynomial of degree } k \text{ and } p(0) = 1.\} \tag{1.23}$$

Theorem 1.2.12. Let A be spd. Then the CG algorithm will find the solution within N iterations.

Proof. Let $\{\lambda_i\}_{i=1}^N$ be the eigenvalues of A. As a test polynomial, let

$$\overline{p}(z) = \prod_{i=1}^{N}(\lambda_i - z)/\lambda_i.$$

$\overline{p} \in \mathcal{P}_N$, hence, by (1.22) and the fact that $\overline{p}$ vanishes on $\sigma(A)$,

$$\|x_N - x^*\|_A \leq \|x_0 - x^*\|_A \max_{z\in\sigma(A)} |\overline{p}(z)| = 0. \qquad \square$$

Remark 1.2.3. This is not as good as it sounds, since in most applications the number of unknowns N is very large. It is best to regard CG as an iterative method.

Theorem 1.2.13. Let A be spd with eigenvectors $\{u_i\}_{i=1}^N$. Let b be a linear combination of k of the eigenvectors of A

$$b = \sum_{l=1}^{k} \gamma_l u_{i_l}.$$

Then the CG iteration for $Ax = b$ with $x_0 = 0$ will terminate in at most k iterations.

Proof. Let $\{\lambda_{i_l}\}$ be the eigenvalues of A associated with the eigenvectors $\{u_{i_l}\}_{l=1}^k$. By the spectral theorem

$$x^* = \sum_{l=1}^{k} (\gamma_l/\lambda_{i_l}) u_{i_l}.$$

We use the residual polynomial,

$$\overline{p}(z) = \prod_{l=1}^{k} (\lambda_{i_l} - z)/\lambda_{i_l}.$$

One can easily verify that $\overline{p} \in \mathcal{P}_k$ and $\overline{p}(\lambda_{i_l}) = 0$ for $1 \leq l \leq k$. So

$$\overline{p}(A)x^* = \sum_{l=1}^{k} \overline{p}(\lambda_{i_l}\gamma_l/\lambda_{i_l}) u_{i_l} = 0.$$

So, we have by (1.18) and the fact that $x_0 = 0$ that

$$\|x_k - x^*\|_A \leq \|\overline{p}(A)x^*\|_A = 0. \qquad \square$$

Theorem 1.2.14. Let A be spd. Assume that there are exactly $k \leq N$ distinct eigenvalues of A. Then the CG iteration terminates in at most k iterations.

1.2.3 Termination of the iteration

Lemma 1.2.15. Let A be spd with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$. Then for all $z \in \mathbb{R}^N$,

$$\|A^{1/2}z\|_2 = \|z\|_A \tag{1.24}$$

and

$$\lambda_N^{1/2}\|z\|_A \leq \|Az\|_2 \leq \lambda_1^{1/2}\|z\|_A. \tag{1.25}$$

Proof.

$$\|z\|_A^2 = z^T Az = (A^{1/2}z)^T(A^{1/2}z) = \|A^{1/2}z\|_2^2$$

Let u_i be a unit eigenvector corresponding to λ_i. We may write $A = U\Lambda U^T$ as

$$Az = \sum_{i=1}^{N} \lambda_i (u_i^T z) u_i.$$

Hence

$$\begin{aligned}\lambda_N\|A^{1/2}z\|_2^2 &= \lambda_N \sum_{i=1}^{N} \lambda_i (u_i^T z)^2 \\ &\leq \|Az\|_2^2 = \sum_{i=1}^{N} \lambda_i^2 (u_i^T z)^2 \\ &\leq \lambda_1 \sum_{i=1}^{N} \lambda_i (u_i^T z)^2 = \lambda_1 \|A^{1/2}z\|_2^2. \qquad \square\end{aligned}$$

Lemma 1.2.16.

$$\frac{\|b - Ax_k\|_2}{\|b\|_2} \leq \frac{\sqrt{\kappa_2(A)}\|r_0\|_2}{\|b\|_2} \frac{\|x_k - x^*\|_A}{\|x^* - x_0\|_A}. \tag{1.26}$$

Proof. Using (1.24) and (1.25) twice, we have

$$\frac{\|b - Ax_k\|_2}{\|b - Ax_0\|_2} = \frac{\|A(x^* - x_k)\|_2}{\|A(x^* - x_0)\|_2} \leq \sqrt{\frac{\lambda_1}{\lambda_N}} \frac{\|x^* - x_k\|_A}{\|x^* - x_0\|_A}. \qquad \square$$

Remark 1.2.4. To predict the performance of the CG iteration based on termination on small relative residuals, we must not only use (1.22) to predict when the relative A-norm error is small, but also use Lemma 1.2.16 to relate small A-norm errors to small relative residuals.

Example 1.2.17. Assume that $x_0 = 0$ and that the eigenvalues of A are contained in the interval $(9, 11)$. If we let $\overline{p}_k(z) = (10 - z)^k/10^k$, then $\overline{p}_k \in \mathcal{P}_k$. This means that we may apply (1.22) to get

$$\|x_k - x^*\|_A \leq \|x^*\|_A \max_{9\leq z\leq 11} |\overline{p}_k(z)|.$$

Hence, after k iterations

$$\|x_k - x^*\|_A \leq \|x^*\|_A 10^{-k}. \tag{1.27}$$

So the size of the A-norm of the error will be reduced by a factor of 10^{-3} when $10^{-k} \leq 10^{-3} \Rightarrow k \geq 3$.

To use Lemma 1.2.16, we simply note that $\kappa_2(A) \leq 11/9$. Hence, after k iterations we have

$$\frac{\|r_k\|_2}{\|b\|_2} \leq \sqrt{11} \times 10^{-k}/3.$$

So, the size of the relative residual will be reduced by a factor of 10^{-3} when $10^{-k} \leq 3 \times 10^{-3}/\sqrt{11} \Rightarrow k \geq 4$.

Example 1.2.18. Assume that $x_0 = 0$ amd the eigenvalues of A lie in the two intervals $(1, 1.5)$ and $(399, 400)$. If we use a residual polynomial $\overline{p}_{3k} \in \mathcal{P}_{3k}$

$$\overline{p}_{3k}(z) = \frac{(1.25 - z)^k (400 - z)^{2k}}{(1.25)^k \times 400^{2k}}.$$

It is easy to see that $\max_{z\in\sigma(A)} |\overline{p}_{3k}(z)| \leq (0.2)^k$, so the size of the A-norm of the error will be reduced by a factor of 10^{-3} when $(0.2)^{-k} \leq 10^{-3} \Rightarrow k \geq 4.3 \Rightarrow k' = 3k \geq 15$.

Theorem 1.2.19.

$$\|x_k - x^*\|_A \leq 2\|x_0 - x^*\|_A \left[\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}\right].$$

Proof. See *The conjugate gradient method for linear and nonlinear operator equations* written by J.W.DANIEL □

Remark 1.2.5. From the theorem 1.2.19 and two examples, we can see that if the condition number of A is near one, the CG itereation will converge very rapidly. What's more, even if the condition number is large, CG can perform very well if the eigenvalues cluster in a few small intervals.

Definition 1.2.20. The transformation of the problem into one with eigenvalues clustered near one (i.e., easier to solve) is called *preconditioning*.

1.2.4 Implementation

Definition 1.2.21. The $(k+1)$th iterate x_{k+1} of CG minimizes $\phi(x)$ over $x_0+\mathcal{K}_{k+1}$, the direction $p_{k+1} \in \mathcal{K}_{k+1}$from x_k to x_{k+1} is called a *search direction* so that $x_{k+1} = x_k + \alpha_{k+1}p_{k+1}$.

Lemma 1.2.22. Once p_{k+1} has been found, we have

$$\alpha_{k+1} = \frac{p_{k+1}^T(b - Ax_k)}{p_{k+1}^T A p_{k+1}} = \frac{p_{l+1}^T r_k}{p_{k+1}^T A p_{k+1}}. \tag{1.28}$$

Proof. x_{k+1} should minimize $\phi(x)$ over $x_0 + \mathcal{K}_{k+1}$, so

$$\frac{\mathrm{d}(x_k + \alpha p_{k+1})}{\mathrm{d}\alpha} = 0 \tag{1.29}$$

for the correct choice of $\alpha = \alpha_{k+1}$. Equation (1.29) can be written as

$$p_{k+1}^T A x_k + \alpha p_{k+1}^T A p_{k+1} - p_{k+1}^T b = 0.$$

So (1.28) is proved. □

Lemma 1.2.23. Let $\{x_k\}$ be the conjugate gradient iterates. Prove that $r_l \in \mathcal{K}_k$ for all $l < k$.

Lemma 1.2.24. Let A be spd and let $\{x_k\}$ be the CG iterates. Then

$$r_k^T r_l = 0 \text{ for all } 0 \leq l < k. \tag{1.30}$$

Proof. Since x_k minimizes ϕ on $x_0 + \mathcal{K}_k$, we have, for any $\xi \in \mathcal{K}_k$,

$$\frac{\mathrm{d}\phi(x_k + t\xi)}{\mathrm{d}t} = \nabla\phi(x_k + t\xi)^T\xi = 0$$

at $t = 0$. Recalling that $\nabla\phi(x) = Ax - b = -r$, we have

$$\nabla\phi(x_k)^T\xi = -r_k^T\xi = 0 \text{ for all } \xi \in \mathcal{K}_k. \tag{1.31}$$

Since $r_l \in \mathcal{K}_k$ for all $l < k$, this proves (1.30). □

Corollary 1.2.25. Let A be spd and let $\{x_k\}$ be the CG iterates. If $x_k = x_{k+1}$, then x_k is the solution.

Proof.

$$\begin{aligned} x_k = x_{k+1} &\Rightarrow r_k = r_{k+1} \\ &\Rightarrow \|r_k\|_2^2 = r_k^T r_k = r_k^T r_{k+1} = 0 \\ &\Rightarrow x_k = x^*. \end{aligned}$$ □

Lemma 1.2.26. Let A be spd and let $\{x_k\}$ be the CG iterates. If $x_k \neq x^*$ then $x_{k+1} = x_k + \alpha_{k+1}p_{k+1}$ and p_{k+1} is determined up to a scalar multiple by the conditions

$$p_{k+1} \in \mathcal{K}_{k+1},\ p_{k+1}^T A\xi = 0 \text{ for all } \xi \in \mathcal{K}_k. \tag{1.32}$$

Proof. Since $\mathcal{K}_k \subset \mathcal{K}_{k+1}$,

$$\nabla\phi(x_{k+1})^T\xi = (Ax_k + \alpha_{k+1}Ap_{k+1} - b)^T\xi = 0 \tag{1.33}$$

for all $\xi \in \mathcal{K}_k$. (1.31) and (1.33) then imply that for all $\xi \in \mathcal{K}_k$,

$$\begin{aligned} \alpha_{k+1}p_{k+1}^T A\xi &= -(Ax_k - b)^T\xi \\ &= -\nabla\phi(x_k)^T\xi = 0. \end{aligned}$$ □

Remark 1.2.6. The condition $p_{k+1}^T A\xi = 0$ is called $A-$*conjugacy* of p_{k+1} to $\mathcal{K}_k$. Now, any p_{k+1} satisfying (1.32) can, up to a scalar multiple, be expressed as

$$p_{k+1} = r_k + \omega_k$$

with $\omega_k \in \mathcal{K}_k$.

Lemma 1.2.27. Let A be spd and assume that $\{d_1, d_1 \ldots, d_n\}$ is $A-$conjugate, then they are linear independent.

Remark 1.2.7. Let A be spd and assume that there is a $A-$conjugate basis of $\mathbb{R}^N$, $\{d_1, d_2, \ldots, d_N\}$, then we have

$$\begin{aligned} \min_{x\in\mathbb{R}^N}\phi(x) &= \min_{a_1,\cdots,a_N\in\mathbb{R}} \frac{1}{2}(\sum_{i=1}^N a_i d_i)^T A(\sum_{j=1}^N a_j d_j) - b^T(\sum_{i=1}^N a_i d_i) \\ &= \min_{a_1,\cdots,a_N\in\mathbb{R}} \frac{1}{2}(\sum_{i=1}^N\sum_{j=1}^N a_i a_j d_i^T A d_j) - \sum_{i=1}^N a_i b^T d_i \\ &= \min_{a_1,\cdots,a_N\in\mathbb{R}} \frac{1}{2}\sum_{i=1}^N (a_i^2 d_i^T A d_i - a_i b^T d_i). \end{aligned}$$

Theorem 1.2.28. Let A be spd and assume that $r_k \neq 0$. Define $p_0 = 0$. Then

$$p_{k+1} = r_k + \beta_{k+1}p_k \text{ for some } \beta_{k+1} \text{ and } k \geq 0. \tag{1.34}$$

Proof. By Lemma 1.2.26 and the fact that $\mathcal{K}_k = \mathbf{span}(r_0, \ldots, r_{k-1})$, we need only verify that a β_{k+1} can be found so that if p_{k+1} is given by (1.34) then

$$p_{k+1}^T A r_l = 0$$

for all $0 \leq l \leq k-1$.

Let p_{k+1} be given by (1.34). Then for any $l < k$

$$p_{k+1}^T A r_l = r_k^T A r_l + \beta_{k+1}p_k^T A r_l.$$

If $l \leq k-2$ then $r_l \in \mathcal{K}_{l+1} \subset \mathcal{K}_{k-1}$. Lemma 1.2.26 implies that $p_{k+1}^T A r_l = 0$ for $0 \leq l \leq k-2$.

It only remains to solve for β_{k+1} so that $p_{k+1}^T A r_{k-1} = 0$. Trivially

$$\beta_{k+1} = -r_k^T A r_{k-1} / p_k^T A r_{k-1} \tag{1.35}$$

provided $p_k^T A r_{k-1} \neq 0$. Since

$$r_k^T r_{k-1} = \|r_{k-1}\|_2^2 - \alpha_k p_k^T A r_{k-1}.$$

Since $r_k^T r_{k-1} = 0$ we have

$$p_k^T A r_{k-1} = \|r_{k-1}\|_2^2 / \alpha_k \neq 0. \tag{1.36}$$

This completes the proof. □

Remark 1.2.8. The common implementation of CG uses a different form for α_k and β_k than given in (1.28) and (1.35).

Lemma 1.2.29. Let A be spd and assume that $r_k \neq 0$. Then

$$\alpha_{k+1} = \frac{\|r_k\|_2^2}{p_{k+1}^T A p_{k+1}} \tag{1.37}$$

and

$$\beta_{k+1} = \frac{\|r_k\|_2^2}{\|r_{k-1}\|_2^2}. \tag{1.38}$$

Proof. Note that for $k \geq 0$

$$p_{k+1}^T r_{k+1} = r_k^T r_{k+1} + \beta_{k+1} p_k^T r_{k+1} = 0 \tag{1.39}$$

by Lemma 1.2.26. An immediate consequence of (1.39) is that $p_k^T r_k = 0$ and hence

$$p_{k+1}^T r_k = (r_k + \beta_{k+1} p_k)^T r_k = \|r_k\|_2^2. \tag{1.40}$$

Taking scalar products of both sides of $r_{k+1} = r_k - \alpha_{k+1} A p_{k+1}$ with p_{k+1} and using (1.40) gives

$$0 = p_{k+1}^T r_k - \alpha_{k+1} p_{k+1}^T A p_{k+1} = \|r_k^T\|_2^2 - \alpha_{k+1} p_{k+1}^T A p_{k+1},$$

which is equivalent to (1.37).

To get (1.38), note that $p_{k+1}^T A p_k = 0$ and hence (1.34) implies that

$$\beta_{k+1} = \frac{-r_k^T A p_k}{p_k^T A p_k}. \tag{1.41}$$

Also note that

$$\begin{aligned} p_k^T A p_k &= p_k^T A (r_{k-1} + \beta_k p_{k-1}) \\ &= p_k^T A r_{k-1} + \beta_k p_k^T A p_{k-1} = p_k^T A r_{k-1}. \end{aligned} \tag{1.42}$$

Now combine (1.41), (1.42) amd (1.36) to get

$$\beta_{k+1} = \frac{-r_k^T A p_k \alpha_k}{\|r_{k-1}\|_2^2}.$$

Now take scalar products of both sides of $r_k = r_{k-1} - \alpha_k A p_k$ with r_k and use Lemma 1.2.26 to get

$$\|r_k\|_2^2 = -\alpha_k r_k^T A p_k.$$

Hence (1.38) holds. □

Algorithm 1.2.30. The usual implementation of conjugate gradient iteration reflects all of the above results.

Algorithm 1: `CG Iteration`

Input: $x \in \mathbb{R}^n$, $b \in \mathbb{R}^n$, $A \in \mathbb{R}^{N \times N}$, $\epsilon \in \mathbb{R}^+$, $k_{\max} \in \mathbb{Z}^+$

Output: The solution which overwrites x, the residual norm $\|r_k\|_2$

Postconditions: $\|r_k\|_2 \leq \epsilon \|b\|_2$ or $k = k_{\max}$

```
r = b − Ax, ρ_0 = ‖r‖_2^2, k = 1;
while √ρ_{k−1} > ε‖b‖_2 and k < k_max do
    if k = 1 then
        p = r;
    else
        β = ρ_{k−1}/ρ_{k−2};
        p = r + βp;
    end
    ω = Ap;
    α = ρ_{k−1}/p^T ω;
    x = x + αp;
    r = r − αω;
    ρ_k = ‖r‖_2^2;
    k = k + 1;
end
```

Remark 1.2.9. The matrix A itself need not be formed or stored, only a routine for matrix-vector products is required. Krylov space methods are often called *matrix-free* for that reason.

Remark 1.2.10. In the CG iteration algorithm, we need store only the four vectors x, ω, p, r. Each iteration requires a single matrix-vector product. two scalar products and three operations of the form $ax + y$.

Remark 1.2.11. CG algorithm can progress without storing a basis for the entire Krylov subspace. The spd structure buys a lot.

1.2.5 Preconditioning

Remark 1.2.12. If M is a spd matrix that is close to A^{-1}, then the eigenvalues of MA will be clustered near one. However, MA is unlikely to be spd.

So we avoid this difficulty by expressing the preconditioned problem in terms of B, where B is spd and $A = B^2$. Then we find a spd approximate inverse of B, S and $M = S^2$. SAS is spd and its eigenvalues are clustered near one. The preconditioned linear system is

$$SASy = Sb$$

which has $y^* = S^{-1}x^*$ as a solution.

Algorithm 1.2.31. If y_k, $\hat{r}_k$, $\hat{p}_k$ are the iterate, residual, and search direction for CG applied to SAS and we let

$$x_k = Sy^k,\ r_k = S^{-1}\hat{r}_k,\ p_k = S\hat{p}_k,\ z_k = S\hat{r}_k$$

then one can perform the iteration directly in terms of x_k, A, M.

Algorithm 2: `PCG Iteration`

Input: $x \in \mathbb{R}^n$, $b \in \mathbb{R}^n$, $A \in \mathbb{R}^{N\times N}$, $M \in \mathbb{R}^{N\times N}, \epsilon \in \mathbb{R}^+, k_{\max} \in \mathbb{Z}^+$

Output: The solution which overwrites x, residual norm $\|r_k\|_2$.

Postconditions: $\|r_k\|_2 \le \epsilon\|b\|_2$ or $k = k_{\max}$

1 $r = b - Ax,\ \rho_0 = \|r\|_2^2,\ k = 1;$
2 **while** $\sqrt{\rho_{k-1}} > \epsilon\|b\|_2$ *and* $k < k_{max}$ **do**
3 $z = Mr;$
4 $\tau_{k-1} = z^T r;$
5 **if** $k = 1$ **then**
6 $\beta = 0;$
7 $p = z;$
8 **else**
9 $\beta = \tau_{k-1}/\tau_{k-2};$
10 $p = z + \beta p;$
11 **end**
12 $\omega = Ap;$
13 $\alpha = \tau_{k-1}/p^T\omega;$
14 $x = x + \alpha p;$
15 $r = r - \alpha\omega;$
16 $\rho_k = r^T r;$
17 $k = k + 1;$
18 **end**

Remark 1.2.13. Note that the cost is identical to CG with the addition of

- the application of the preconditioner M in line 3.
- the additional inner product required to compute τ_k in line 4.

1.2.6 Examples for preconditioned conjugate iteration

Example 1.2.32. We consider the discretization of the PDE

$$-\nabla\cdot(a(x,y)\nabla u) = f(x,y) \tag{1.43}$$

on $0 < x, y < 1$ subject to homogeneous Dirichlet boundary conditions

$$u(x,0) = u(x,1) = u(0,y) = u(1,y) = 0, \quad 0 < x, y < 1.$$

One can verify that the five-point discretization is positive definite if $a > 0$ for all $0 \ge x, y \ge 1$.

For the computations we take $a(x,y) = \cos(x)$ and take the right hand side so that the exact solution is the discretization of

$$10xy(1-x)(1-y)\exp(x^{4.5}).$$

The initial iterate is $u_0 = \mathbf{0}$.

In the results reported here we take $n = 31$ resulting in a system with $N = n^2 = 961$ unknowns. We expect second-order accuracy from the method and accordingly we set termination parameter $\epsilon = h^2 = 1/1024$. We allowed up to 100 CG iterations. The initial iterate is the zero vector. For a preconditioner in PCG, we use a Poisson solver. We will report our results graphically.

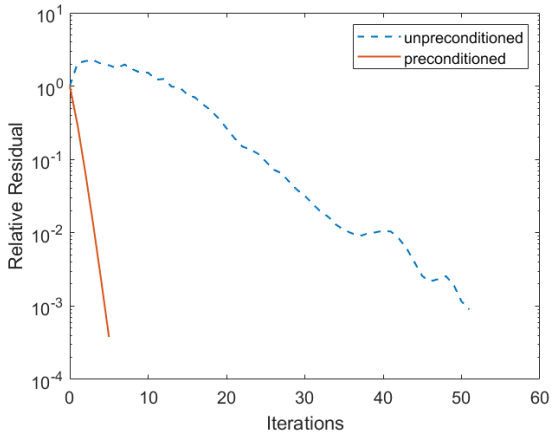

Figure 1.1: relative residual of CG and PCG

In Figure 1.1 the solid line is a plot of $\|r_k\|_2/\|b\|_2$ for the preconditioned iteration and the dashed line for the unpreconditioned. Note that the unpreconditioned reduction in $\|r\|$ is not monotone. This is consistent with the theory, which predicts decrese in $\|e\|_A$ but not necessarily in $\|r\|$ as the iteration progresses.

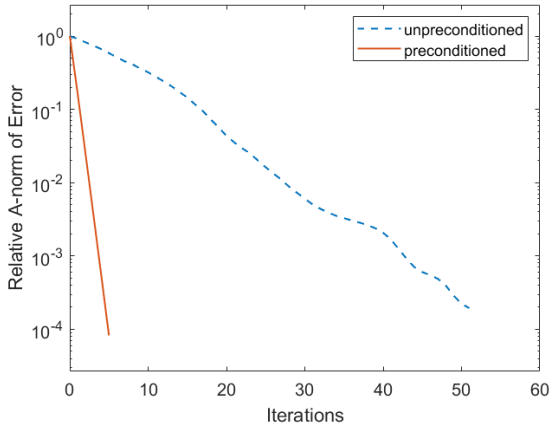

Figure 1.2: relative error of CG and PCG

In Figure 1.2 the solid line is a plot of $\|u^* - u_k\|_A/\|u^* - u_0\|_A$ for the preconditioned iteration and the dashed line for the unpreconditioned. The preconditioned iteration required 5 iterations for convergence and the unpreconditioned iteration 52. Note that the unpreconditioned iteration is slowly convergent. This can be explained by the fact that the eigenvalues are not clustered and

$$\kappa(A) = O(1/h^2) = O(n^2) = O(N).$$

1.2.7 CGNR and CGNE

Definition 1.2.33. If A is nonsingular and nonsymmetric, one might consider solving $Ax = b$ by applying CG to the normal equations

$$A^T Ax = A^T b. \tag{1.44}$$

This approach is called CGNR.

Remark 1.2.14. The reason for this name is that the minimization property of CG as applied to (1.45) asserts that

$$\begin{aligned}\|x^* - x\|^2_{A^TA} &= (x^* - x)^T A^T A(x^* - x)\\ &= (Ax^* - Ax)^T(Ax^* - Ax) = \|r\|_2^2\end{aligned}$$

is minimized over $x_0+\mathcal{K}_k$ at each iterate. Hence it is called CG on the Normal equations to minimize the Residual.

Definition 1.2.34. If A is nonsingular and nonsymmetric, one could also consider applying CG to the normal equations

$$AA^Ty = b \tag{1.45}$$

and then set $x = A^Ty$. This approach is called CGNE.

Remark 1.2.15. The reason for this name is that the minimization property of CG as applied to (1.45) asserts that

$$\begin{aligned}\|y^* - y\|^2_{AA^T} &= (y^* - y)^T AA^T(y^* - y)\\ &= (A^Ty^* - A^Ty)^T(A^Ty^* - A^Ty)\\ &= \|x^* - x\|_2^2\end{aligned}$$

is minimized over $y_0+\mathcal{K}_k$ at each iterate. Hence it is called CG on the Normal equations to minimize the Error.

Remark 1.2.16. There are three disadvantages that may or may not be serious:

- The condition number of A^TA is the square of that of A.
- Two matrix-vector products are needed for each CG iterate since $\omega = A^T(Ap)$ in CGNR and $\omega = A(A^Tp)$ in CGNE.
- One must compute the action of A^T on a vector as part of the matrix-vector product involving A^TA.

1.3 GMRES Iteration

1.3.1 The minimization property and its consequences

Theorem 1.3.1 (The minimization property of GMRES iteration)**.** Let A be nonsingular. The kth iteration of GMRES is the solution to the least squares problem

$$\text{minimize}_{x\in x_0+\mathcal{K}_k}\|b - Ax\|_2. \tag{1.46}$$

Then for all $\overline{p}_k \in \mathcal{P}_k$

$$\|r_k\|_2 = \min_{p\in\mathcal{P}_k}\|p(A)r_0\|_2 \le \|\overline{p}_k(A)r_0\|_2 \tag{1.47}$$

Proof. Note that if $x \in x_0 + \mathcal{K}_k$, then

$$x = x_0 + \sum_{j=0}^{k-1}\gamma_j A^j r_0$$

and so

$$b - Ax = b - Ax_0 - \sum_{j=0}^{k-1}\gamma_j A^{j+1}r_0 = r_0 - \sum_{j=1}^{k}\gamma_{j-1}A^j r_0$$

Hence if $x \in x_0 + \mathcal{K}_k$ then $r = \overline{p}(A)r_0$ where $\overline{p} \in \mathcal{P}_k$ is a residual polynomial, so (1.47) is proved. □

Corollary 1.3.2. Let A be nonsingular and let x_k be the kth GMRES iteration. Then for all $\overline{p}_k \in \mathcal{P}_k$

$$\frac{\|r_k\|_2}{\|r_0\|_2} \le \|\overline{p}_k(A)\|_2. \tag{1.48}$$

Theorem 1.3.3. Let A be nonsingular. Then the GMRES algorithm will find the solution within N iterations.

Proof. The characteristic polynomial of A is $p(z) = \det(A - zI)$. p has degree N and $p(0) = \det(A) \neq 0$ since A is nonsingular, so

$$\overline{p}_N(z) = p(z)/p(0) \in \mathcal{P}_N$$

is a residual polynomial. It is well known that $p(A) = 0 = \overline{p}_N(A)$. By (1.48), $r_N = b - Ax_N = 0$ and hence x_N is the solution. □

Theorem 1.3.4. Let $A = V\Lambda V^{-1}$ be a nonsingular diagonalizable matrix. Let x_k be the kth GMRES iterate. Then for all $\overline{p}_k \in \mathcal{P}_k$

$$\frac{\|r_k\|_2}{\|r_0\|_2} \le \kappa_2(V)\max_{z\in\sigma(A)}|\overline{p}_k(z)|. \tag{1.49}$$

Proof. Let $\overline{p}_k \in \mathcal{P}_k$. We can easily estimate $\|\overline{p}_k(A)\|_2$ by

$$\|\overline{p}_k(A)\|_2 \le \|V\|_2\|V^{-1}\|_2\|\overline{p}_k(\Lambda)\|_2 \le \kappa_2(V)\max_{z\in\sigma(A)}|\overline{p}_k(z)|,$$

as asserted. □

Corollary 1.3.5. If A is normal, then we have

$$\frac{\|r_k\|_2}{\|r_0\|_2} \le \max_{z\in\sigma(A)}|\overline{p}_k(z)|.$$

Theorem 1.3.6. Let A be a nonsingular diagonalizable matrix. Assume that A has only k distinct eigenvalues. Then GMRES will terminate in at most k iterations.

Theorem 1.3.7. Let A be a nonsingular normal matrix. Let b be a linear combination of k of the eigenvectors of A

$$b = \sum_{l=1}^{k}\gamma_l u_{i_l}.$$

Then the GMRES iteration, with $x_0 = 0$, for $Ax = b$ will terminate in at most k iterations.

1.3.2 Termination

Example 1.3.8. Assume that $A = V\Lambda V^{-1}$ is diagonalizable, that the eigenvalues of A lie in the interval $(9, 11)$, and that $\kappa_2(V) = 100$. We assume that $x_0 = 0$ and hence $r_0 = b$. Using residual polynomial $\overline{p}_k(z) = (10 - z)^k/10^k$ we find

$$\frac{\|r_k\|_2}{\|r_0\|_2} \le (100)10^{-k} = 10^{2-k}.$$

Hence

$$\|r_k\|_2 \le \eta\|b\|_2 \tag{1.50}$$

holds when $k > 2 + \log_{10}(\eta)$.

Lemma 1.3.9. Assume that A is nonsingular, diagonalizable and $\|I - A\|_2 \leq \rho < 1$. Let $\overline{p}_k(z) = (1-z)^k$. We have

$$\|r_k\|_2 \leq \rho^k \|r_0\|_2. \tag{1.51}$$

Remark 1.3.1. The estimate (1.51) illustrates the potential benefits of a good approximate inverse precondition.

1.3.3 Preconditioning

Definition 1.3.10. If one can find M such that

$$\|I - MA\|_2 = \rho < 1,$$

then applying GMRES to $MAx = Mb$ allows one to apply (1.51) to the preconditioned system. Preconditioning done in this way is called *left preconditioning.*

Remark 1.3.2. If $r = MAx - Mb$ allows one to apply (1.51) to the preconditioned system, we have

$$\frac{\|e_k\|_2}{\|e_0\|_2} \leq \kappa_2(MA) \frac{\|r_k\|_2}{\|r_0\|_2}.$$

If MA has a smaller condition number than A, we might expect the relative residual of the preconditioned system to be a better indicator of the relative error than that of the original system.

Definition 1.3.11. If one can find M such that

$$\|I - AM\|_2 = \rho < 1,$$

one can solve $AMy = b$ with GMRES and then set $x = My$. This is called *right preconditioning.*

Remark 1.3.3. The residual of the preconditioned problem is the same as that of the unpreconditioned problem. Right preconditioning has been used as the basis for a method that changes the preconditioner as the iteration progresses.

1.3.4 Implementation: Basic ideas

Lemma 1.3.12. The least squares problem defining the kth GMRES iterate,

$$\text{minimize}_{x \in x_0 + \mathcal{K}_k} \|b - Ax\|_2,$$

is equivalent to the least squares problem in $\mathbb{R}^k$,

$$\text{minimize}_{y \in \mathbb{R}^k} \|r_0 - AV_k y\|_2, \tag{1.52}$$

where V_k is an orthogonal projector onto $\mathcal{K}_k$.

Proof. For any $z \in \mathcal{K}_k$, z can be written as

$$z = \sum_{l=1}^{k} y_l v_l^k,$$

where v_l^k is the lth column of V_k. Hence we can convert (1.46) to a least squares problem in $\mathbb{R}^k$ for the coefficient vector y of $z = x - x_0$. Since $x - x_0 = V_k y$ for some $y \in \mathbb{R}^k$, we must have $x_k = x_0 + V_k y$ where y minimizes

$$\|b - A(x_0 + V_k y)\|_2 = \|r_0 - AV_k y\|_2. \qquad \square$$

Remark 1.3.4. We could solve this problem with a QR factorization. However, the problem is that the matrix vector product of A with the basis matrix V_k must be taken at each iteration.

Algorithm 1.3.13. The Gram-Schmit procedure for formation of an orthonormal basis for $\mathcal{K}_k$ is called the *Arnoldi process.*

Algorithm 3: `Arnodi Process`

Input: $x_0 \in \mathbb{R}^n$, $b \in \mathbb{R}^n$, $A \in \mathbb{R}^{N \times N}$, $k \in \mathbb{Z}^+$
Output: Orthonormal basis for $\mathcal{K}_k$ stored in V

$r_0 = b - Ax_0$, $v_1 = r_0 / \|r_0\|_2$;
for $i = 1, \ldots, k-1$ **do**
  $v_{i+1} = \frac{Av_i - \sum_{j=1}^{i}((Av_i)^T v_j)v_j}{\|Av_i - \sum_{j=1}^{i}((Av_i)^T v_j)v_j\|_2}$
end

Lemma 1.3.14. Let A be nonsingular, let the vectors v_j be generated by Algorithm `Arnoldi`, and let i be the smallest integer for which

$$Av_i - \sum_{j=1}^{i}((Av_i)^T v_j)v_j = 0.$$

Then $x = A^{-1}b \in x_0 + \mathcal{K}_i$.

Proof. By hypothesis $Av_i \in \mathcal{K}_i$ and hence $A\mathcal{K}_i \subset \mathcal{K}_i$. Since the columns of V_i are an orthonormal basis for $\mathcal{K}_i$, we have

$$AV_i = V_i H,$$

where H is an $i \times i$ matrix. H is nonsingular since A is. Setting $\beta = \|r_0\|_2$ and $e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^i$, we have

$$\|r_i\|_2 = \|b - Ax_i\|_2 = \|r_0 - A(x_i - x_0)\|_2.$$

Since $x_i - x_0 \in \mathcal{K}_i$ there is $y \in \mathbb{R}^i$ such that $x_i - x_0 = V_i y$. Since $r_0 = \beta V_i e_1$ and V_i is an orthogonal matrix

$$\|r_i\|_2 = \|V_i(\beta e_1 - Hy)\|_2 = \|\beta e_1 - Hy\|_{\mathbb{R}^{i+1}},$$

where $\|\cdot\|_{\mathbb{R}^{i+1}}$ denotes the Euclidean norm in $\mathbb{R}^{i+1}$.

Setting $y = \beta H^{-1} e_1$ proves that $r_i = 0$ by the minimization property. $\square$

Lemma 1.3.15. The `Arnoldi process`(unless it terminates permaturely with a solution) produces matrices $\{V_k\}$ with orthonormal columns such that there exists an upper Hessenberg matrix $H_k \in \mathbb{R}^{(k+1)\times k}$,

$$AV_k = V_{k+1} H_k \tag{1.53}$$

Proof. Set $h_{ij} = (Av_j)^T v_i$, it is easy to prove that H_k is upper Hessenberg and (1.53) holds. $\square$

Corollary 1.3.16. The least squares problem in the kth GMRES iterate is equivalent to the least squares problem in $\mathbb{R}^k$

$$\text{minimize}_{y^k \in \mathbb{R}^k} \|\beta e_1 - H_k y^k\|_{\mathbb{R}^{k+1}}.$$

Proof. For some $y^k \in \mathbb{R}^k$,

$$r_k = b - Ax_k = r_0 - A(x_k - x_0) = V_{k+1}(\beta e_1 - H_k y^k).$$

Hence $x_k = x_0 + V_k y^k$, where y^k minimizes $\|\beta e_1 - H_k y\|_2$ over $\mathbb{R}^k$.

Note that when y^k has been computed, the norm of r_k can be found without explicitly forming x_k and computing $b - Ax_k$. We have

$$\begin{aligned}\|r_k\|_2 &= \|V_{k+1}(\beta e_1 - H_k y^k)\|_2 \\ &= \|\beta e_1 - H_k y^k\|_{\mathbb{R}^{k+1}}.\end{aligned}$$ □

Algorithm 1.3.17. The usual implementation of GMRES iteration reflects all of the above results.

Algorithm 4: `GMRESa`

Input: $x \in \mathbb{R}^n$, $b \in \mathbb{R}^n$, $A \in \mathbb{R}^{N\times N}$, $\epsilon \in \mathbb{R}^+, k_{\max} \in \mathbb{Z}^+$
Output: The solution which overwrites x, the residual norm ρ.
Postconditions: $\rho \le \epsilon\|b\|_2$ or $k = k_{\max}$

1 $r = b - Ax,\ v_1 = r/\|r\|_2,\ \rho = \|r\|_2^2$;
2 $k = 0,\ \beta = \rho$;
3 **while** $\rho > \epsilon\|b\|_2$ *and* $k < k_{max}$ **do**
4 $k = k + 1$;
5 **for** $j = 1, \ldots, k$ **do**
6 $h_{jk} = (Av_k)^T v_j$;
7 **end**
8 $v_{k+1} = Av_k - \sum_{j=1}^k h_{jk} v_j$;
9 $h_{k+1,k} = \|v_{k+1}\|_2$;
10 $v_{k+1} = v_{k+1}/\|v_{k+1}\|_2$;
11 $e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^{k+1}$;
12 Minimize $\|\beta e_1 - H_k y^k\|_{\mathbb{R}^{k+1}}$ over $\mathbb{R}^k$ to obtain y^k;
13 $\rho = \|\beta e_1 - H_k y^k\|_{\mathbb{R}^{k+1}}$;
14 **end**
15 $x_k = x_0 + V_k y^k$;

Remark 1.3.5. Note that x_k is only computed upon termination and is not needed within the iteration. It is an important property of GMRES that the basis for the Krylov space must be stored as the iteration progress.

Definition 1.3.18. For very large problems, one way to avoid the problem in Remark 1.3.5 is to set $k_{\max}$ to the maximum number m of vectors that one can store, call GMRES and explicitly test the residual $b - Ax_k$ if $k = m$ upon termination. If the norm of the residual is larger than ϵ, call GMRES again with $x_0 = x_k$.

The restarted version of the algorithm is called *GMRES(m)*.

Example 1.3.19. Let $\delta = 10^{-7}$ and define

$$A = \begin{pmatrix} 1 & 1 & 1 \\ \delta & \delta & 0 \\ \delta & 0 & \delta \end{pmatrix}.$$

We orthogonalize the columns of A with classical Gram-Schmidt to obtain

$$V = \begin{pmatrix} 1.0000e+00 & 1.0436e-07 & 9.9715e-08 \\ 1.0000e-07 & 1.0456e-14 & -9.9905e-01 \\ 1.0000e-07 & -1.0000e+00 & 4.3568e-02 \end{pmatrix}.$$

The columns of V are not orthogonal at all. In fact $v_2^T v_3 \approx -0.004$.

Algorithm 1.3.20. A partial remedy is to replace the classical Gram-Schmidt orthogonalization in Algorithm `GMRESa` with modified Gram-Schmidt orthogonalization.

Algorithm 5: `GMRESb`

Input: $x \in \mathbb{R}^n$, $b \in \mathbb{R}^n$, $A \in \mathbb{R}^{N\times N}$, $\epsilon \in \mathbb{R}^+, k_{\max} \in \mathbb{Z}^+$
Output: The solution which overwrites x, the residual norm ρ.
Postconditions: $\rho \le \epsilon\|b\|_2$ or $k = k_{\max}$

1 $r = b - Ax,\ v_1 = r/\|r\|_2,\ \rho = \|r\|_2^2$;
2 $k = 0,\ \beta = \rho$;
3 **while** $\rho > \epsilon\|b\|_2$ *and* $k < k_{max}$ **do**
4 $k = k + 1$;
5 $v_{k+1} = Av_k$;
6 **for** $j = 1, \ldots, k$ **do**
7 $h_{jk} = (v_{k+1})^T v_j$;
8 $v_{k+1} = v_{k+1} - h_{jk} v_j$;
9 **end**
10 $h_{k+1,k} = \|v_{k+1}\|_2$;
11 $v_{k+1} = v_{k+1}/\|v_{k+1}\|_2$;
12 $e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^{k+1}$;
13 Minimize $\|\beta e_1 - H_k y^k\|_{\mathbb{R}^{k+1}}$ over $\mathbb{R}^k$ to obtain y^k;
14 $\rho = \|\beta e_1 - H_k y^k\|_{\mathbb{R}^{k+1}}$;
15 **end**
16 $x_k = x_0 + V_k y^k$;

Example 1.3.21. We take the same matrix A as that in Example 1.3.19, for modified Gram-Schmidt, we have

$$V = \begin{pmatrix} 1.0000e+00 & 1.0436e-07 & 1.0436e-07 \\ 1.0000e-07 & 1.0456e-14 & -1.0000e+00 \\ 1.0000e-07 & -1.0000e+00 & 4.3565e-16 \end{pmatrix}.$$

Hence $|v_i^T v_j - \delta| \le 10^{-8}$ for all i, j.

Remark 1.3.6. Even if modified Gram-Schmidt orthogonalization is used, one can still lose orthogonality in the columns of V. Reorthogonalization is necessary if A is ill conditioned. One easy way is to augment the modified Gram-Schmidt process with a second pass. The implementation of reorthogonalization will be given in the end of the section.

Remark 1.3.7. The kth GMRES iteration requires a matrix-vector product, k scalar products, and the solution of the Hessenberg least problem in line 13. If we solve this problem by QR factorization, then the total cost of the m GMRES iterations is $O(m^4)$.

1.3.5 Implementatin: Givens rotations

Definition 1.3.22. A 2×2 *Givens rotation* is a matrix of the form

$$G = \begin{pmatrix} c & -s \\ s & c \end{pmatrix}, \tag{1.54}$$

where $c = \cos(\theta),\ s = \sin(\theta)$ for $\theta \in [-\pi, \pi]$.

Lemma 1.3.23. The 2×2 Givens rotation rotates the vector $(c, -s)$, which makes an angle of $-\theta$ with the x-axis through an angle θ so that it overlaps the x-axis.

$$G \begin{pmatrix} c \\ -s \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Definition 1.3.24. An $N \times N$ Givens rotation replaces a 2×2 block on the diagonal of the $N \times N$ identity matrix with a 2×2 Givens rotation.

$$G = \begin{pmatrix} 1 & 0 & & \cdots & & & 0 \\ 0 & \ddots & \ddots & & & & \\ & \ddots & c & -s & & & \\ \vdots & & s & c & 0 & & \vdots \\ & & & 0 & 1 & \ddots & \\ & & & & \ddots & \ddots & 0 \\ & & & \cdots & & 0 & 1 \end{pmatrix} \tag{1.55}$$

$G_j(c, s)$ is an $N \times N$ Givens rotation of the form (1.55) with a 2×2 Givens rotation in rows and columns j and $j + 1$.

Lemma 1.3.25. Let H be an $N \times M$ $(N \geq M)$ upper Hessenberg matrix with rank M. There exists a product of Givens rotations Q such that $QH = R$ is upper triangular.

Proof. We reduce H to triangular form by first multiplying the matrix by a Givens rotation that annihilates h_{21} (and of course, changes h_{11} and the subsequent columns). We define $G_1 = G_1(c_1, s_1)$ by

$$c_1 = h_{11}/\sqrt{h_{11}^2 + h_{21}^2} \text{ and } s_1 = -h_{21}/\sqrt{h_{11}^2 + h_{21}^2}. \tag{1.56}$$

If we replace H by $G_1 H$, then the first column of H now has only a single nonzero element h_{11}. Similary we can now apply $G_2(c_2, s_2)$ to H, where

$$c_2 = h_{22}/\sqrt{h_{22}^2 + h_{32}^2} \text{ and } s_2 = -h_{32}/\sqrt{h_{22}^2 + h_{32}^2}. \tag{1.57}$$

and annihilate h_{32}. Note that G_2 does not affect the first column of H. Continuing in this way and setting

$$Q = G_N \dots G_1,$$

we see that $QH = R$ is upper triangular. □

Lemma 1.3.26. The least squares problem

$$\text{minimize}_{y^k \in \mathbb{R}^k} \|\beta e_1 - H_k y^k\|_{\mathbb{R}^{k+1}}$$

is equivalent to the least squares problem in $\mathbb{R}^k$

$$\text{minimize}_{y^k \in \mathbb{R}^k} \|g - R_k y^k\|_{\mathbb{R}^{k+1}}$$

where R_k is the $k + 1 \times k$ triangular factor of the QR factorization of H_k and $g = \beta Q e_1$.

Algorithm 1.3.27. The complete GMRES iteration reflects all of the above results.

Algorithm 6: `GMRES`

Input: $x \in \mathbb{R}^n$, $b \in \mathbb{R}^n$, $A \in \mathbb{R}^{N \times N}$,$\delta \in \mathbb{R}^+$, $\epsilon \in \mathbb{R}^+, k_{\max} \in \mathbb{Z}^+$

Output: The solution which overwrites x, the residual norm ρ.

Postconditions: $\rho \leq \epsilon \|b\|_2$ or $k = k_{\max}$

$r = b - Ax,\ v_1 = r/\|r\|_2,\ \rho = \|r\|_2^2$;
$k = 0,\ \beta = \rho$;
$g = \rho(1, 0, \dots, 0)^T \in \mathbb{R}^{k_{\max}+1}$;
while $\rho > \epsilon \|b\|_2$ *and* $k < k_{max}$ **do**
  $k = k + 1$;
  $v_{k+1} = Av_k$;
  for $j = 1, \dots, k$ **do**
    $h_{jk} = (v_{k+1})^T v_j$;
    $v_{k+1} = v_{k+1} - h_{jk} v_j$;
  end
  if $\|Av_k\|_2 + \delta \|v_{k+1}\|_2 = \|Av_k\|_2$ **then**
    for $j = 1, \dots, k$ **do**
      $h_{tmp} = (v_{k+1})^T v_j$;
      $h_{jk} = h_{jk} + h_{tmp}$;
      $v_{k+1} = v_{k+1} - h_{tmp} v_j$;
    end
  end
  $h_{k+1,k} = \|v_{k+1}\|_2$;
  $v_{k+1} = v_{k+1}/\|v_{k+1}\|_2$;
  if $k > 1$ **then**
    apply Q_{k-1} to the kth column of H;
  end
  $\nu = \sqrt{h_{k,k}^2 + h_{k+1,k}^2}$;
  $c_k = h_{k,k}/\nu,\ s_k = -h_{k+1,k}/\nu$;
  $h_{k,k} = c_k h_{k,k} - s_k h_{k+1,k},\ h_{k+1,k} = 0$;
  $g = G_k(c_k, s_k) g$;
  $\rho = |(g)_{k+1}|$;
end
Set $r_{i,j} = h_{i,j}$ for $1 \leq i, j \leq k$;
Set $(\omega)_i = (g)_i$ for $1 \leq i \leq k$;
Solve the upper triangular system $Ry^k = \omega$;
$x_k = x_0 + V_k y^k$;

Remark 1.3.8. The cost of one single GMRES iteration is $O(N)$ floating-point operations. The $O(N^2)$ cost of the triangular solve and the $O(N)$ cost of the construction of x_k are incurred after termination. Hence the total cost of the m GMRES iterations is $O(m^2)$.

1.3.6 Examples for GMRES iteration

Example 1.3.28. We consider the discretization of the PDE

$$\begin{aligned}(Lu)(x, y) = &-(u_{xx}(x, y) + u_{yy}(x, y)) + a_1(x, y) u_x(x, y) \\ &+ a_2(x, y) u_y(x, y) + a_3(x, y) u(x, y) = f(x, y)\end{aligned} \tag{1.58}$$

on $0 < x, y < 1$ subject to homogeneous Dirichlet boundary conditions

$$u(x,0) = u(x,1) = u(0,y) = u(1,y) = 0, \quad 0 < x, y < 1.$$

For the computations reported in this section we took

$$a_1(x,y) = 1, \ a_2(x,y) = 20y, \ a_3(x,y) = 1.$$

As before we discretize with a five-point centered difference scheme with n^2 points and mesh width $h = 1/(n+1)$. Then let $n = 31$ to create a system with 961 unknowns. As a preconditioner we use the fast Poisson solver, let Gu denote the action of the Poisson solver on u, the preconditioned system is $GLu = Gf$.

We take $u_0 = 0$ be the initial iterate and the right hand side so that the exact solution is the discretization of

$$10xy(1-x)(1-y)\exp(x^{4.5})$$

In Figure 1.3 we plot iteration histories corresponding to preconditioned and unpreconditioned GMRES.

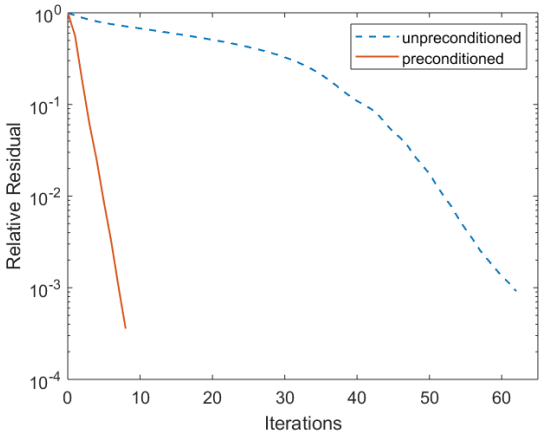

Figure 1.3: relative residual of PGMRES and GMRES.